

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2001 年 11 月 1 日 (01.11.2001)

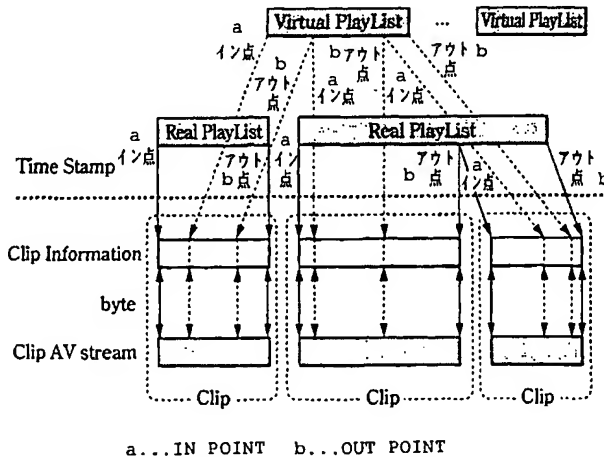
PCT

(10) 国際公開番号
WO 01/82606 A1

- (51) 国際特許分類: H04N 5/92, G11B 20/10 Motoki [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).
- (21) 国際出願番号: PCT/JP01/03415
- (22) 国際出願日: 2001 年 4 月 20 日 (20.04.2001) (74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (81) 指定国 (国内): AU, BR, CA, CN, CZ, HU, ID, IL, IN, KR, MX, NO, NZ, PL, RO, RU, SG, SK, US, VN, ZA.
- (30) 優先権データ: (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
- 特願2000-183771 2000 年 4 月 21 日 (21.04.2000) JP
特願2000-271552 2000 年 9 月 7 日 (07.09.2000) JP
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).
- 添付公開書類:
— 国際調査報告書
— 補正書
- (72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 加藤元樹 (KATO, 2 文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体



(57) Abstract: A CPI-type is described in a Playlist(). The CPI-type includes an EP-map type and a TU-map type. If the position of an I picture can be analyzed, the EP-map type is used; and if not, the TU-map type is used. As a result, AV stream data recorded after analyzing the position of the I picture and AV stream data recorded without analyzing the position of the I picture can be commonly managed.

(57) 要約:

Playlist()には、CPI_typeが記述される。CPI_typeには、EP_maptpeと、TU_m ap typeがある。I ピクチャの位置が分析できる場合、EP_maptpeが用いられ、I ピクチャの位置が分析できない場合、TU_map typeが用いられる。これにより、I ピクチャの位置を分析してAVストリームデータと、分析しないで記録するAVストリームデータとを共通に管理できるようにする。

BEST AVAILABLE COPY

WO 01/82606 A1

明細書

情報処理装置及び方法、プログラム、並びに記録媒体

技術分野

本発明は情報処理装置、再生方法、プログラム、並びに記録媒体に関し、特に、G U I 等に説明表示する情報、主の再生経路の情報、副の再生経路の情報、主の再生経路を構成する個々の再生区間の間の接続情報、ユーザが所望したシーンにセットするブックマークやリジューム点の情報等の情報を含むファイルを記録する情報処理装置、再生方法、プログラム、並びに記録媒体に関する。

背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のA V (Audio Visual) 信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのA V 信号のソース（供給源）としては、C S デジタル衛星放送やB S デジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるデジタルビデオ信号は、通常M P E G (Moving Picture Experts Group) 2 方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。或いは、M P E G 1 Video、M P E G 2 Video、D V 方式をはじめとするデジタル記録方式であれば、1 度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。但し、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要がある。

また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

ところで、上述したような記録装置により記録媒体にAVストリームデータを記録する場合、例えば、高速再生ができるようにするために、Aストリームデータを分析し、Iピクチャの位置を検出して、Iピクチャにアクセスできるようにして記録する場合と、AVストリームデータを分析せず、そのまま記録する場合とがある。

このような場合、従来、それぞれ専用のアプリケーションプログラムを用意し、それぞれにより、AVストリームを、異なるフォーマットのAVストリーム（高速再生が可能なAVストリーム、又は不可能なAVストリーム）として記録媒体に記録するようにしていた。この結果、アプリケーションプログラムの開発に、費用と時間がかかる課題があった。また、それぞれのアプリケーションプログラムにより記録されたAVストリームは、異なるフォーマットのものとなので、相

互の互換性がなくなり、共通の装置で再生することができなくなる課題があった。

更に、従来の記録装置では、例えば、オーディオデータを、所謂アフターレコーディングすることが困難である課題があった。

発明の開示

本発明の目的は、このような状況に鑑み、高速再生が可能なAVストリームと不可能なAVストリームを、共通に管理することにある。

また、本発明の他の目的は、アフターレコーディングを可能にすることにある。

本発明に係る情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する第1の生成手段と、記録方法に応じて第1のテーブル又は第2のテーブルの一方を選択する選択手段と、選択されたテーブルをAVストリームデータとともに記録媒体に記録する第1の記録手段とを有する。

前記第1のテーブルは、EP_mapであり、第2のテーブルは、TU_mapとすることができる。

前記選択手段は、ノンコグニザント記録の際には、第2のテーブルを選択することができる。

前記選択手段は、セルフエンコード記録の際には、第1のテーブルを選択することができる。

前記選択手段は、コグニザント記録の際には、第1のテーブルを選択することができる。

前記AVストリームデータの再生を指定する再生指定情報を生成する第2の生成手段と、第2の生成手段により生成された再生指定情報を記録媒体に記録する第2の記録手段を更に有し、再生指定情報は、AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現するか、又はアライバル

タイムベースで表現するかを示す種別情報を含むようにすることができる。

前記ＡＶストリームデータとともに第１のテーブルが記録されている場合、再生指定情報は、ＡＶストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現し、ＡＶストリームデータとともに第２のテーブルが記録されている場合、再生指定情報は、ＡＶストリームデータの再生区間の時間情報を、アライバルタイムベースで表現することができる。

本発明に係る情報処理方法は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのＡＶストリームデータ中のアドレスとの対応関係を記述する第１のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのＡＶストリームデータ中のアドレスとの対応関係を記述する第２のテーブルを生成する生成ステップと、記録方法に応じて第１のテーブル又は第２のテーブルの一方を選択する選択ステップと、選択されたテーブルをＡＶストリームデータとともに記録媒体に記録する記録ステップとを含む。

本発明に係る記録媒体のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのＡＶストリームデータ中のアドレスとの対応関係を記述する第１のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのＡＶストリームデータ中のアドレスとの対応関係を記述する第２のテーブルを生成する生成ステップと、記録方法に応じて第１のテーブル又は第２のテーブルの一方を選択する選択ステップと、選択されたテーブルをＡＶストリームデータと共に記録媒体に記録する記録ステップとを含む。

本発明に係るプログラムは、プレゼンテーションタイムスタンプと、これに対応するアクセスユニットのＡＶストリームデータ中のアドレスとの対応関係を記述する第１のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのＡＶストリームデータ中のアドレスとの対応関係を記述する第２のテーブルを生成する生成ステップと、記録方法に応じて第１のテーブル又は第２のテーブルの一方を選択する選択ステップと、選択されたテーブルをＡＶストリームデータと共に記録

媒体に記録する記録ステップとをコンピュータに実行させる。

本発明に係る情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのA Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのA Vストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生手段と、再生されたテーブルに基づいて、A Vストリームデータの出力を制御する制御手段とを有する。

本発明に係る情報処理方法は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのA Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのA Vストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、A Vストリームデータの出力を制御する制御ステップとを含む。

本発明に係る記録媒体のプログラムは、プレゼンテーションタイムスタンプと、これに対応するアクセスユニットのA Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのA Vストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、A Vストリームデータの出力を制御する制御ステップとを含む。

本発明に係るプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのA Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのA Vスト

リームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている。

本発明に係る情報処理装置は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成手段と、AVストリームデータと再生指定情報を記録媒体に記録する記録手段とを備える。

前記副の再生パスは、オーディオデータのアフターレコーディング用のパスとすることができる。

前記第1の情報は、Main_pathであり、第2の情報は、Sub_pathとすることができる。

前記第2の情報は、副の再生パスのタイプを表すタイプ情報、副の再生パスが参照するAVストリームのファイル名、副の再生パスのAVストリームのイン点とアウト点、及び再生パスのイン点が、主のパスの時間軸上で同期してスタートする主のパス上の時刻を含むようにすることができる。

本発明に係る情報処理方法は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとを含む。

本発明に係る記録媒体のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情

報を記録媒体に記録する記録ステップとを含む。

本発明に係るプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、A Vストリームデータと再生指定情報を記録媒体に記録する記録ステップとをコンピュータに実行させる。

前記情報処理装置は、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生手段と、再生された前記再生指定情報に基づいて、前記A Vストリームデータの出力を制御する制御手段とを備える。

本発明に係る情報処理方法は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、A Vストリームデータの出力を制御する制御ステップとを含む。

本発明に係る記録媒体のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、A Vストリームデータの出力を制御する制御ステップとを含む。

本発明に係るプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、A Vストリームデータの出力を制御する制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録されている。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、プログラム、並びに記録媒体においては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記A Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたア

ライバルタイムスタンプと、それに対応するトランスポートパケットの前記A Vストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムにおいては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記A Vストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記A Vストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、そのテーブルが再生され、それに基づいて、出力が制御される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、プログラム、並びに第2の記録媒体においては、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムにおいては、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録媒体から再生され、それに基づいて、出力が制御される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の構成を示す図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlaylistとVirtual Playlistについて説明する図である。

図4A～図4Cは、Real Playlistの作成について説明する図である。

図 5 A～図 5 C は、Real Playlist の削除について説明する図である。

図 6 A 及び図 6 B は、アセンブル編集について説明する図である。

図 7 は、Virtual Playlist にサブパスを設ける場合について説明する図である。

図 8 は、Playlist の再生順序の変更について説明する図である。

図 9 は、Playlist 上のマークと Clip 上のマークについて説明する図である。

図 10 は、メニューサムネイルについて説明する図である。

図 11 は、Playlist に付加されるマークについて説明する図である。

図 12 は、クリップに付加されるマークについて説明する図である。

図 13 は、Playlist、Clip、サムネイルファイルの関係について説明する図である。

図 14 は、ディレクトリ構造について説明する図である。

図 15 は、info.dvr のシンタクスを示す図である。

図 16 は、DVR volume のシンタクスを示す図である。

図 17 は、Resume volume のシンタクスを示す図である。

図 18 は、UIAppInfo volume のシンタクスを示す図である。

図 19 は、Character set value のテーブルを示す図である。

図 20 は、TableOfPlaylist のシンタクスを示す図である。

図 21 は、TableOfPlaylist の他のシンタクスを示す図である。

図 22 は、MakersPrivateData のシンタクスを示す図である。

図 23 は、xxxxx.rpls と yyyyy.vpls のシンタクスを示す図である。

図 24 A～図 24 C は、Playlist について説明する図である。

図 25 は、Playlist のシンタクスを示す図である。

図 26 は、Playlist_type のテーブルを示す図である。

図 27 は、UIAppinfoPlaylist のシンタクスを示す図である。

図 28 A～図 28 C は、図 27 に示した UIAppinfoPlaylist のシンタクス内のフラグについて説明する図である。

図 29 は、PlayItem について説明する図である。

図 30 は、PlayItem について説明する図である。

図 31 は、PlayItem について説明する図である。

- 図 3 2 は、PlayItemのシンタクスを示す図である。
- 図 3 3 は、IN_timeについて説明する図である。
- 図 3 4 は、OUT_timeについて説明する図である。
- 図 3 5 は、Connection_Conditionのテーブルを示す図である。
- 図 3 6 A～図 3 6 D は、Connection_Conditionについて説明する図である。
- 図 3 7 は、BridgeSequenceInfoを説明する図である。
- 図 3 8 は、BridgeSequenceInfoのシンタクスを示す図である。
- 図 3 9 は、SubPlayItemについて説明する図である。
- 図 4 0 は、SubPlayItemのシンタクスを示す図である。
- 図 4 1 は、SubPath_typeのテーブルを示す図である。
- 図 4 2 は、PlaylistMarkのシンタクスを示す図である。
- 図 4 3 は、Mark_typeのテーブルを示す図である。
- 図 4 4 は、Mark_time_stampを説明する図である。
- 図 4 5 は、zzzzz.clipのシンタクスを示す図である。
- 図 4 6 は、ClipInfoのシンタクスを示す図である。
- 図 4 7 は、Clip_stream_typeのテーブルを示す図である。
- 図 4 8 は、offset_SPNについて説明する図である。
- 図 4 9 は、offset_SPNについて説明する図である。
- 図 5 0 A 及び図 5 0 B は、S T C 区間について説明する図である。
- 図 5 1 は、STC_Infoについて説明する図である。
- 図 5 2 は、STC_Infoのシンタクスを示す図である。
- 図 5 3 は、ProgramInfoを説明する図である。
- 図 5 4 は、ProgramInfoのシンタクスを示す図である。
- 図 5 5 は、VideoCondngInfoのシンタクスを示す図である。
- 図 5 6 は、Video_formatのテーブルを示す図である。
- 図 5 7 は、frame_rateのテーブルを示す図である。
- 図 5 8 は、display_aspect_ratioのテーブルを示す図である。
- 図 5 9 は、AudioCondngInfoのシンタクスを示す図である。
- 図 6 0 は、audio_codingのテーブルを示す図である。

- 図 6 1 は、audio_component_type のテーブルを示す図である。
- 図 6 2 は、sampling_frequency のテーブルを示す図である。
- 図 6 3 は、CPI について説明する図である。
- 図 6 4 は、CPI について説明する図である。
- 図 6 5 は、CPI のシンタクスを示す図である。
- 図 6 6 は、CPI_type のテーブルを示す図である。
- 図 6 7 は、ビデオ EP_map について説明する図である。
- 図 6 8 は、EP_map について説明する図である。
- 図 6 9 は、EP_map について説明する図である。
- 図 7 0 は、EP_map のシンタクスを示す図である。
- 図 7 1 は、EP_type values のテーブルを示す図である。
- 図 7 2 は、EP_map_for_one_stream_PID のシンタクスを示す図である。
- 図 7 3 は、TU_map について説明する図である。
- 図 7 4 は、TU_map のシンタクスを示す図である。
- 図 7 5 は、ClipMark のシンタクスを示す図である。
- 図 7 6 は、mark_type のテーブルを示す図である。
- 図 7 7 は、mark_type_stamp のテーブルを示す図である。
- 図 7 8 は、menu.thmb と mark.thmb のシンタクスを示す図である。
- 図 7 9 は、Thumbnail のシンタクスを示す図である。
- 図 8 0 は、thumbnail_picture_format のテーブルを示す図である。
- 図 8 1 A 及び図 8 1 B は、tn_block について説明する図である。
- 図 8 2 は、DVR MPEG 2 のトランスポートストリームの構造について説明する図である。
- 図 8 3 は、DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。
- 図 8 4 は、DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。
- 図 8 5 は、source packet のシンタクスを示す図である。
- 図 8 6 は、TP_extra_header のシンタクスを示す図である。

図 8 7 は、copy permission indicatorのテーブルを示す図である。

図 8 8 は、シームレス接続について説明する図である。

図 8 9 は、シームレス接続について説明する図である。

図 9 0 は、シームレス接続について説明する図である

図 9 1 は、シームレス接続について説明する図である。

図 9 2 は、シームレス接続について説明する図である

図 9 3 は、オーディオのオーバーラップについて説明する図である。

図 9 4 は、BridgeSequenceを用いたシームレス接続について説明する図である。

図 9 5 は、BridgeSequenceを用いないシームレス接続について説明する図である。

図 9 6 は、DVR STDモデルを示す図である。

図 9 7 は、復号、表示のタイミングチャートである。

図 9 8 は、PlayListファイルのシンタクスを示す図である。

図 9 9 は、図 9 8 のPlayListファイル中のUIAppInfoPlayListのシンタクスを示す図である。

図 1 0 0 は、図 9 8 のPlayListファイル中のPlayList()のシンタクスを示す図である。

図 1 0 1 は、SubPlayItemのシンタクスを示す図である。

図 1 0 2 は、Real PlayListの作成方法を説明するフローチャートである。

図 1 0 3 は、Virtual PlayListの作成方法を説明するフローチャートである。

図 1 0 4 は、PlayListの再生方法を説明するフローチャートである。

図 1 0 5 は、PlayListのSubバスの再生方法を説明するフローチャートである。

図 1 0 6 は、PlayListMarkの作成方法を説明するフローチャートである。

図 1 0 7 は、PlayListMarkを使用した頭出し再生方法を説明するフローチャートである。

図 1 0 8 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された情報処理装置及び方法、プログラム、並びに記録媒体について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置1は、アナログデータ、又は、デジタルデータを入力し、記録することができる構成とされている。

端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、AVエンコーダ15に出力される。解析部14は、入力されたビデオ信号からシーンチェンジ等の特徴点を抽出する。

AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、及びAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、このストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC(誤り訂正)符号化部20、変調部21で所定の処理が施され、書込部22に出力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

ディジタルインタフェース又はディジタルテレビジョンチューナから入力されるディジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、これらは、トランスペアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子24から制御部23へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、及びシステム情報(S)を抽出する。

デマルチプレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、及び、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記

録する場合と同一の処理なので、その説明は省略する。

記録再生装置 1 は、A V ストリームのファイルを記録媒体 1 0 0 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部 2 3 により作成される。制御部 2 3 への入力情報は、解析部 1 4 からの動画像の特徴情報、多重化ストリーム解析部 1 8 からの A V ストリームの特徴情報、及び端子 2 4 から入力されるユーザからの指示情報である。

解析部 1 4 から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル (C M) の開始・終了点等の指定情報 (マーク) であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

多重化ストリーム解析部 1 8 からの A V ストリームの特徴情報は、記録される A V ストリームの符号化情報に関する情報であり、例えば、A V ストリーム内の I ピクチャのアドレス情報、A V ストリームの符号化パラメータ、A V ストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関する情報 (マーク) 等である。

端子 2 4 からのユーザの指示情報は、A V ストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報等である。

制御部 2 3 は、上記の入力情報に基づいて、A V ストリームのデータベース (Clip)、A V ストリームの再生区間 (PlayItem) をグループ化したもの (PlayList) のデータベース、記録媒体 1 0 0 の記録内容の管理情報 (info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、A V ストリームと同様にして、E C C 符号化部 2 0、変調部 2 1 で処理されて、書込部 2 2 へ入力される。書込部 2 2 は、制御部 2 3 から出力される制御信号に基づいて、記録媒体 1 0 0 へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体 1 0 0 に記録された A V ストリームファイル (画像データと音声データのファイル) と、アプリケーションデータベース情報が再生さ

れる場合、先ず、制御部 23 は、読出部 28 に対して、記録媒体 100 からアプリケーションデータベース情報を読み出すように指示する。そして、読出部 28 は、記録媒体 100 からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部 29、ECC 復号部 30 の処理を経て、制御部 23 へ入力される。

制御部 23 は、アプリケーションデータベース情報に基づいて、記録媒体 100 に記録されている PlayList の一覧を端子 24 のユーザインタフェースへ出力する。ユーザは、PlayList の一覧から再生したい PlayList を選択し、再生を指定された PlayList に関する情報が制御部 23 へ入力される。制御部 23 は、その PlayList の再生に必要な AV ストリームファイルの読み出しを、読出部 28 に指示する。読出部 28 は、その指示に従い、記録媒体 100 から対応する AV ストリームを読み出し復調部 29 に出力する。復調部 29 に入力された AV ストリームは、所定の処理が施されることにより復調され、更に ECC 復号部 30 の処理を経て、ソースデパケッタ 31 へ出力される。

ソースデパケッタ 31 は、記録媒体 100 から読み出され、所定の処理が施されたアプリケーションフォーマットの AV ストリームを、デマルチプレクサ 26 に出力できるストリームに変換する。デマルチプレクサ 26 は、制御部 23 により指定された AV ストリームの再生区間(PlayItem)を構成するビデオストリーム (V)、オーディオストリーム (A)、及び AV 同期等のシステム情報 (S) を、AV デコーダ 27 に出力する。AV デコーダ 27 は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子 32 と端子 33 から出力する。

また、ユーザインタフェースとしての端子 24 から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部 23 は、AV ストリームのデータベース(Clip)の内容に基づいて、記憶媒体 100 からの AV ストリームの読出位置を決定し、その AV ストリームの読み出しを、読出部 28 に指示する。例えば、ユーザにより選択された PlayList を、所定の時刻から再生する場合、制御部 23 は、指定された時刻に最も近いタイムスタンプを持つ I ピクチャからのデータを読み出すように読出部 28 に指示する。

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読出部28に指示する。

読出部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書込部22に指示する。

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

先ず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読出部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読出部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケットイザ31を経て、デマル

チプレクサ 2 6 に出力される。

制御部 2 3 は、デマルチプレクサ 2 6 に入力されたデータを解析して、ビデオストリームの再エンコード方法 (picture_coding_type の変更、再エンコードする符号化ビット量の割り当て) と、再多重化方式を決定し、その方式を A V エンコーダ 1 5 とマルチプレクサ 1 6 に供給する。

次に、デマルチプレクサ 2 6 は、入力されたストリームをビデオストリーム (V)、オーディオストリーム (A)、及びシステム情報 (S) に分離する。ビデオストリームは、「A V デコーダ 2 7 に入力されるデータ」と「マルチプレクサ 1 6 に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これは A V デコーダ 2 7 で復号され、復号されたピクチャは A V エンコーダ 1 5 で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ 1 6 に入力される。

マルチプレクサ 1 6 は、制御部 2 3 から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、E C C 符号化部 2 0、変調部 2 1 で処理されて、書込部 2 2 に入力される。書込部 2 2 は、制御部 2 3 から供給される制御信号に基づいて、記録媒体 1 0 0 に A V ストリームを記録する。

以下、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図 2 は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、A V ストリームの管理のために Playlist と Clip の 2 つのレイヤをもつ。Volume Information は、ディスク内の全ての Clip と Playlist の管理をする。ここでは、1 つの A V ストリームとその付属情報のペアを 1 つのオブジェクトとし、これを Clip という。A V ストリームファイルは Clip AV stream file といい、この付属情報は、Clip Information file という。

1 つの Clip AV stream file は、M P E G 2 トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。

一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、これを簡単に編集することができるようにするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4Aは、Real PlayListのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4Bは、Real PlayListのディバイド(divide:分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

図 4 Cは、Real PlayListのコンバイン(combine: 結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される(Clip自体が1つにされる)ことはない。

図 5 Aは、Real PlayList全体のデリート(delete: 削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図 5 Bは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図 5 Cは、Real PlayListのミニマイズ(Minimize: 最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayList にとって不必要なClipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか?」といったメッセージ等を表示させることにより、確認(警告)を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図 6 A及び

図 6 B は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと

所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

図 6 A に示したように、2つのReal PlayList 1, 2と、それぞれのReal Play Listに対応するClip 1, 2が存在している場合に、ユーザがReal PlayList 1内の所定の区間(In 1乃至Out 1までの区間: PlayItem 1)を再生区間として指示し、続けて再生する区間として、Real PlayList 2内の所定の区間(In 2乃至Out 2までの区間: PlayItem 2)を再生区間として指示したとき、図 6 B に示すように、PlayItem 1とPlayItem 2から構成される1つのVirtual PlayListが作成される。

次に、Virtual PlayList の再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加(append)、Virtual PlayListの中のPlayItemの削除等がある。また、Virtual PlayListそのものを削除することもできる。

図 7 は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブバスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインバスのAVストリームに、付加的なオーディオストリームが、サブバスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図 8 に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList (図 2 0 等を参照して後述する) によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク (Mark) について説明する。マークは、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加される

マークは、A Vストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点等である。PlayListを再生するとき、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点等である。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

次に、サムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）等を実行して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットしたときに、そのディスクの内容を表す静止画を最初に表示する場合等に用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコ

ンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル (Mark Thumbnail s) という。したがって、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図 1 1 は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図 1 2 は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表すときに、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置 1 がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは 1 つのファイルに格納するのがよい。Playlistはメニューサムネイル 1 つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図 1 3 は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、C P I (Characteristic Point Information) について説明する。C P I は、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のC P Iを用いる。1つは、EP_mapであり、

もう1つは、TU_mapである。

EP_mapは、エントリポイント（EP）データのリストであり、これは、エレメンタリーストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリーム

ファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置（図1の記録再生装置1）にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリーストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMP EG-2 AACのオーディオストリーム等）等の情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MP EG 2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリーストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。

ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceという。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

ここでは、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号（例えばDV）をデコードしてからMP EG 2トランスポートストリームに符号化する場合に用いられる。

SESFは、MP EG-2トランスポートストリーム及びAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、

S E S Fストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをS E S Fストリームにトランスコーディングする。この場合、記録されたストリームは、S E S Fに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがI S D B（日本のデジタルB S放送の規格名称）準拠のM P E G - 2トランスポートストリームであり、それがH D T VビデオストリームとM P E G A A Cオーディオストリームを含むとする。H D T VビデオストリームをS D T Vビデオストリームにトランスコーディングし、そのS D T VビデオストリームとオリジナルのA A CオーディオストリームをT Sに再多重化する。S D T Vストリームと記録されるトランスポートストリームは、共にI S D Bフォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにEP_mapが作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにTU_mapが作成されてディスクに記録される。

次にディレクトリとファイルについて説明する。以下、記録再生装置1をD V R（Digital Video Recording）と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。D V Rのディスク上に必要なディレクトリは、

図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、及び“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、それらは、本例のアプリケーションフォーマットでは、無視されとする。

“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、デジタルTV放送等のデータ放送のファイルがストアされる。

“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されとする。“menu.thmb”ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

“mark.thmb”ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

“PLAYLIST”ディレクトリは、2種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。“xxxxx.rpls” ファイ

ルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal Play List毎に、1つのファイルが作られる。ファイル名は、“xxxxx.rpls”である。ここで、“xxxxx”は、5個の0乃至9まで数字である。ファイル拡張子は、“rpls”でなければならないとする。

“yyyyy.vpls”ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、“yyyyy.vpls”である。ここで、“yyyyy”は、5個の0乃至9まで数字である。ファイル拡張子は、“vpls”でなければならないとする。

“CLIPINF”ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。“zzzzz.clpi”ファイルは、1つのAVストリームファイル(Clip AV stream file 又は Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、“zzzzz.clpi”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“clpi”でなければならないとする。

“M2TS”ディレクトリは、AVストリームのファイルをストアする。“zzzzz.m2ts”ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、“zzzzz.m2ts”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“m2ts”でなければならないとする。

“DATA”ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイル等である。

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、“info.dvr”ファイルについて説明する。図15は、“info.dvr”ファイルのシンタクスを示す図である。“info.dvr”ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、及びMakerPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明すると、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウン

トされる。

MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、info.dvrのシンタクスに従って挿入される。N 1とN 2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値をとるようにしてもよい。

DVRVolume()は、ボリューム (ディスク) の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVR Volume()のシンタクスを説明すると、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

ResumeVolume()は、ボリュームの中で最後に再生したReal PlayList又はVirtual PlayListのファイル名を記憶している。但し、Real PlayList又はVirtual PlayListの再生をユーザが中断したときの再生位置は、PlayListMark()において定義されるresume-markにストアされる。

図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明すると、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlayList又はVirtual PlayListのファイル名を示す。

図16に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolume は、ボリュームについてのユーザインタフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明すると、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方

法は、図 1 9 に示される値に対応する。

name_lengthの 8 ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが 1 にセットされている場合、ユーザが正しくP I N番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが 0 にセットされている場合、ユーザがP I N番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが 0 にセットされているか、又は、このフラグが 1 にセットされていてもユーザがP I N番号を正しく入力できたならば、記録再生装置 1 は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

P I Nは、4 個の 0 乃至 9 までの数字で構成され、それぞれの数字は、I S O / I E C 6 4 6 に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に、図 1 5 に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されている全てのPlayListファ

イルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明すると、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの10バイトの数字は、PlayListのファイル名を示す。

図21は、TableOfPlayLists()のシンタクスを他の例の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlayList(後述)を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、これを定義したメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでもよい。

所定のメーカーが、プライベートデータを挿入したいときに、既に他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施例では、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれ

ることが可能であるようにする。

図 2 2 は、MakersPrivateDataのシンタクスを示す図である。図 2 2 に示したMakersPrivateDataのシンタクスについて説明すると、version_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数は0からカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカープライベートデータのエントリ数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

maker_model_codeは、そのメーカープライベートデータを作成したDVRシステムのモデルナンバコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。start_mpd_block_numberは、そのメーカープライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカープライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカープライベートデータの大きさを示す32ビ

ットの符号なし整数である。mpd_blockは、メーカープライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real PlayList)、又は、yyyyy.vpls (Virtual PlayList) のシンタクスを示す図である。xxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、PlayList()、PlayListMark()、及びMakerPrivateData()である。

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding_word (パディングワード) は、PlayListファイルのシンタクスにしたがって挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値をとるようにしてもよい。

ここで、既に、簡便に説明したが、PlayListについて更に説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip (後述) を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24Aから図24Cを参照して更に説明すると、図24Aに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、何れかのReal PlayListを参照することにより、必ず視聴することが可能である。

図24Cに示したように、Virtual PlayListの再生区間は、Real PlayListの再

生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

RealPlaylistは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual Playlistは、PlayItemのリストを含み、Playlist()の中に示されるCPI_typeがEP_map typeであり、且つPlaylist_typeが0（ビデオとオーディオを含むPlaylist）である場合、Virtual Playlistは、1つのSubPlayItemを含むことができる。Playlist()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual Playlistが持つSubPlayItemの数は、0又は1でなければならない。

次に、Playlistについて説明する。図25は、Playlistのシンタクスを示す図である。図25に示したPlaylistのシンタクスを説明すると、version_numberは、このPlaylist()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlaylist()の最後まで Playlist()のバイト数を示す32ビットの符号なし整数である。Playlist_typeは、このPlaylistのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

CPI_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlaylistによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、Playlistの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、Playlistの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームのパス（オーディオストリームパス）は、サブパスの一種である。

次に、図25に示したPlaylistのシンタクスのUIAppInfoPlaylistについて説明する。UIAppInfoPlaylistは、Playlistについてのユーザインタフェースアプリケ

ーションのパラメータをストアする。図 27 は、UIAppInfoPlayListのシンタクスを示す図である。図 27 に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、8 ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図 19 に示したテーブルに準拠する値に対応する。

name_lengthは、8 ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録されたときの日時をストアする 56 ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、14 個の数字を 4 ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

durationは、PlayListの総再生時間を時間／分／秒の単位で示した 24 ビットのフィールドである。このフィールドは、6 個の数字を 4 ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

valid_periodは、PlayListが有効である期間を示す 32 ビットのフィールドである。このフィールドは、8 個の数字を 4 ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置 1 は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置 1）の製造者を示す 16 ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す 16 ビットの

符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

write_protect_flagは、図28Aにテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、そのReal PlayListを"Minimize"する。

is_played_flagは、図28Bに示すように、フラグが1にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

archiveは、図28Cに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、

0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次に、PlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、及び、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

PlayListが2つ以上のPlayItemから構成されるとき、これらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一系列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持たないとき、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じS T C連続区間上の時間を指していなければならない。このような例を図29に示す。

図30は、PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持つとき、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time (図の中でIN_time1と示されているものは、先行するPlayItemのSTC_sequence_idによって指定されるS T C連続区間上の時間を指している。先行するPlayItemのOUT_time (図の中でOUT_time1と示されているものは、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time (図の中でIN_time2と示されているものは、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time (図の中でOUT_time2と示されているものは、現在のPlayItemのSTC_sequence_idによって指定されるS T C連続区間上の時間を指している。

図 3 1 に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AV ストリーム上の時間を指している。

PlayItemのシンタクスは、図 3 2 に示すようになる。図 3 2 に示したPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

STC_sequence_idは、8 ビットのフィールドであり、PlayItemが参照するSTC 連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この8 ビットフィールドは何も意味を持たず、0 にセットされる。IN_timeは、32 ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図 3 3 に示すように、PlayList()において定義されるCPI_typeによって異なる。

OUT_timeは、32 ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図 3 4 に示すように、PlayList()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図 3 5 に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2 ビットのフィールドである。図 3 6 A～図 3 6 D は、図 3 5 に示したConnection_Conditionの各状態について説明する図である。

次に、BridgeSequenceInfoについて、図 3 7 を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。更に現在のPlayItemが参照するClip AV stream上のソースパケッ

トのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipという。

図 3 7 において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

図 3 8 は、BridgeSequenceinfoのシンタクスを示す図である。図 3 8 に示したBridgeSequenceinfoのシンタクスを説明すると、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの 3 2 ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの 3 2 ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図 3 9 を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合に許される。ここでは、

SubPlayItemはオーディオのアフレコの目的のためだけに使用されとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、S T C不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値

の上位32ビットを示す。

$$\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$$

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明すると、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、

それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル) は、図45に示すように6個のオブジェクトから構成される。これらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、及びMakerPrivateData()である。AVストリーム(Clip AVストリーム又はBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の”zzzzz”が使用される。

図45に示したzzzzz.clpi (Clip information fileファイル) のシンタクスについて説明すると、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数は0からカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数は0からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData ()の先頭アドレ

スを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N1, N2, N3, N4、及びN5は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリーム又はBridge-Clip AVストリームファイル) の属性情報をストアする。

図46に示したClipInfoのシンタクスについて説明すると、version_numberは、このClipInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリーム又はBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録されるとき、このoffset_SPNは0でなければならない。

図48に示すように、AVストリームファイルのはじめの部分が編集によって消去されたとき、offset_SPNは、0以外の値をとってもよい。ここでは、offset_SPNを参照する相対ソースパケット番号 (相対アドレス) が、しばしば、RSPN_xxx (xxxは変形する。例、RSPN_EP_start) の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN_xxx) は、次式で算出される。

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

図 4 8 に、offset_SPN が 4 である場合の例を示す。

TS_recording_rate は、24 ビットの符号なし整数であり、この値は、DVR ドライブ（書込部 22）へ又は DVR ドライブ（読出部 28）からの AV ストリームの必要な入出力のビットレートを与える。record_time_and_date は、Clip に対応する AV ストリームが記録されたときの日時をストアする 56 ビットのフィールドであり、年／月／日／時／分／秒について、14 個の数字を 4 ビットの Binary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203” と符号化される。

duration は、Clip の総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した 24 ビットのフィールドである。このフィールドは、6 個の数字を 4 ビットの Binary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30 は、“0x014530” と符号化される。

time_controlled_flag: のフラグは、AV ストリームファイルの記録モードを示す。この time_controlled_flag が 1 である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\begin{aligned} \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) - \alpha &\leq \text{size_clip}(t) \\ &\leq \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha \end{aligned}$$

ここで、TS_average_rate は、AV ストリームファイルのトランスポートストリートの平均ビットレートを bytes/second の単位で表したものである。

また、上式において、t は、秒単位で表される時間を示し、start_time は、AV ストリームファイルの最初のソースパケットが記録されたときの時刻であり、秒単位で表される。size_clip(t) は、時刻 t における AV ストリームファイルのサイズをバイト単位で表したものであり、例えば、start_time から時刻 t までに 10 個のソースパケットが記録された場合、size_clip(t) は 10*192 バイトである。 α は、TS_average_rate に依存する定数である。

time_controlled_flag が 0 にセットされている場合、記録モードは、記録の時間経過と AV ストリームのファイルサイズが比例するように制御していないことを

示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。先ずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

に基づいて算出される。

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1であるとき、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_service_ID_validのフラグが1である場合、service_IDのフィールドが有効であることを示す。

is_country_code_validのフラグが1であるとき、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration_descriptor (ISO/IEC13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビ

ットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

service_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservice_IDの値を示す。country_codeの24ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

format_identifier、original_network_ID、transport_stream_ID、service_ID、country_code、及びstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点(システムタイムベースの不連続点)を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない(但し、後述するように、Clipの最大時間長は制限されている)。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続(システムタイムベースの不連続)が発生する場

所のアドレスをストアする。図 5 1 を参照して説明するように、RSPN_STC_start が、そのアドレスを示し、最後の STC_sequence を除く k 番目 ($k \geq 0$) の STC_sequence は、 k 番目の RSPN_STC_start で参照されるソースパケットが到着した時刻から始まり、 $(k + 1)$ 番目の RSPN_STC_start で参照されるソースパケットが到着した時刻で終わる。最後の STC_sequence は、最後の RSPN_STC_start で参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図 5 2 は、STC_Info のシンタクスを示す図である。図 5 2 に示した STC_Info のシンタクスについて説明すると、version_number は、この STC_Info() のバージョンナンバを示す 4 個のキャラクター文字である。version_number は、ISO 646 に従って、"0045" と符号化されなければならない。

length は、この length フィールドの直後から STC_Info() の最後まで STC_Info() のバイト数を示す 32 ビットの符号なし整数である。CPI() の CPI_type が TU_map type を示す場合、この length フィールドは 0 をセットしてもよい。CPI() の CPI_type が EP_map type を示す場合、num_of_STC_sequences は 1 以上の値でなければならない。

num_of_STC_sequences の 8 ビットの符号なし整数は、Clip 中での STC_sequence の数を示す。この値は、このフィールドに続く for-loop のループ回数を示す。所定の STC_sequence に対応する STC_sequence_id は、RSPN_STC_start を含む for-loop の中で、その STC_sequence に対応する RSPN_STC_start の現れる順番により定義されるものである。STC_sequence_id は、0 から開始される。

RSPN_STC_start の 32 ビットフィールドは、AV ストリームファイル上で STC_sequence が開始するアドレスを示す。RSPN_STC_start は、AV ストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_start は、AV ストリームの中で新しいシステムタイムベースの最初の PCR を持つソースパケットの相対アドレスとしてもよい。RSPN_STC_start は、ソースパケット番号を単位とする大きさであり、AV ストリームファイルの最初のソースパケットから ClipInfo() において定義される offset_SPN の値を初期値としてカウントされる。その AV stream ファイルの中での絶対アドレスは、既に上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。

次に、図 4 5 に示した zzzzz.clip のシンタクス内の ProgramInfo について説明する。図 5 3 を参照しながら説明すると、ここでは、Clip の中で次の特徴をもつ時間区間を program_sequence と呼ぶ。まず、PCR_PID の値が変わらない。次に、ビデオエレメンタリーストリームの数が増減しない。また、それぞれのビデオストリームについての PID の値とその VideoCodingInfo によって定義される符号化情報が増減しない。更に、オーディオエレメンタリーストリームの数が増減しない。また、それぞれのオーディオストリームについての PID の値とその AudioCodingInfo によって定義される符号化情報が増減しない。

program_sequence は、同一の時刻において、ただ 1 つのシステムタイムベースを持つ。program_sequence は、同一の時刻において、ただ 1 つの PMT を持つ。ProgramInfo() は、program_sequence が開始する場所のアドレスをストアする。RSPN_program_sequence_start が、そのアドレスを示す。

図 5 4 は、ProgramInfo のシンタクスを示す図である。図 5 4 に示した ProgramInfo のシンタクスを説明すると、version_number は、この ProgramInfo() のバージョンナンバを示す 4 個のキャラクター文字である。version_number は、ISO 646 に従って、"0045" と符号化されなければならない。

length は、この length フィールドの直後から ProgramInfo() の最後までの ProgramInfo() のバイト数を示す 32 ビットの符号なし整数である。CPI() の CPI_type が TU_map type を示す場合、この length フィールドは 0 にセットされてもよい。CPI() の CPI_type が EP_map type を示す場合、number_of_programs は 1 以上の値でなければならない。

number_of_program_sequences の 8 ビットの符号なし整数は、Clip の中で program_sequence の数を示す。この値は、このフィールドに続く for-loop のループ回数を示す。Clip の中で program_sequence が変化しない場合、number_of_program_sequences は 1 をセットされなければならない。RSPN_program_sequence_start の 32 ビットフィールドは、AV ストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明すると、video_formatの8ビットフィールドは、図56に示すように、ProgramInf

o()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明すると、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、これらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_map又は1つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明すると、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイ

ブを表す。

次に、図 6 5 に示した C P I のシンタクス内の EP_map について説明する。EP_map には、2 つのタイプがあり、それはビデオストリーム用の EP_map とオーディオストリーム用の EP_map である。EP_map 中の EP_map_type が、EP_map のタイプを区別する。Clip が 1 つ以上のビデオストリームを含む場合、ビデオストリーム用の EP_map が使用されなければならない。Clip がビデオストリームを含まず、1 つ以上のオーディオストリームを含む場合、オーディオストリーム用の EP_map が使用されなければならない。

ビデオストリーム用の EP_map について図 6 7 を参照して説明する。ビデオストリーム用の EP_map は、stream_PID、PTS_EP_start、及び、RSPN_EP_start というデータを持つ。stream_PID は、ビデオストリームを伝送するトランスポートパケットの P I D を示す。PTS_EP_start は、ビデオストリームのシーケンスヘッダから始めるアクセスユニットの P T S を示す。RSPN_EP_start は、A V ストリームの中で PTS_EP_start により参照されるアクセスユニットの第 1 バイト目を含むソースポケットのアドレスを示す。

EP_map_for_one_stream_PID() と呼ばれるサブテーブルは、同じ P I D を持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clip の中に複数のビデオストリームが存在する場合、EP_map は複数の EP_map_for_one_stream_PID() を含んでもよい。

オーディオストリーム用の EP_map は、stream_PID、PTS_EP_start、及び RSPN_EP_start というデータを持つ。stream_PID は、オーディオストリームを伝送するトランスポートパケットの P I D を示す。PTS_EP_start は、オーディオストリームのアクセスユニットの P T S を示す。RSPN_EP_start は、A V ストリームの中で PTS_EP_start で参照されるアクセスユニットの第 1 バイト目を含むソースポケットのアドレスを示す。

EP_map_for_one_stream_PID() と呼ばれるサブテーブルは、同じ P I D を持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clip の中に複数のオーディオストリームが存在する場合、EP_map は複数の EP_map_for_one_stream_PID() を含んでもよい。

EP_mapとSTC_Infoの関係を説明すると、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる(図68を参照)。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明すると、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリストリーム(ビデオ又はオーディオストリーム)を伝送するトランスポートパケットのPIDを示す。EP_typeが0('video')に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

padding_wordは、EP_map()のシンタクスに従って挿入されなければならない。
XとYは、0 又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値をとってもよい。

図 7 2 は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図 7 2 に示したEP_map_for_one_stream_PIDのシンタクスを説明すると、PTS_EP_startの 32 ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの 33 ビット精度のPTSの上位 32 ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの 33 ビット精度のPTSの上位 32 ビットを持つ。

RSPN_EP_startの 32 ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースボケットの相対アドレスを示す。又は、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースボケットの相対アドレスを示す。

RSPN_EP_startは、ソースボケット番号を単位とする大きさであり、AVストリームファイルの最初のソースボケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図 7 3 を参照して説明する。TU_mapは、ソースボケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、0

ffset_timeから一定の単位に分割される。その単位を、time_unitという。

A Vストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのA Vストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startという。TU_map_time_axis上において、k (k >=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$TU_start_time(k) = offset_time + k * time_unit_size$$

TU_start_time(k)は、45 MHzの精度を持つ。

図75は、TU_mapのシンタクスを示す図である。図75に示したTU_mapのシンタクスを説明すると、offset_timeの32ビット長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27 MHz精度のアライバルタイムクロックから導き出される45 kHzクロックを単位とする大きさである。A Vストリームが新しいClipとして記録される場合、offset_timeは0にセットされなければならない。

time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27 MHz精度のアライバルタイムクロックから導き出される45 MHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size <= 45000) にすることがよい。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリ数を示す。

RSPN_time_unit_startの32ビットフィールドは、A Vストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$SPN_xxx = RSPN_xxx - offset_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k + 1)番目のtime_unitの中にソースパケット

が何もない場合、 $(k + 1)$ 番目のRSPN_time_unit_startは、 k 番目のRSPN_time_unit_startと等しくなければならない。

図 4 5 に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。Clip Markは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置 1）によってセットされるものであり、ユーザによってセットされるものではない。

図 7 5 は、ClipMarkのシンタクスを示す図である。図 7 5 に示したClipMarkのシンタクスを説明すると、version_numberは、このClipMark()のバージョンナンバを示す 4 個のキャラクター文字である。version_numberは、ISO 646 に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す 32 ビットの符号なし整数である。number_of_clip_marksは、ClipMarkの中にストアされているマークの個数を示す 16 ビットの符号なし整数であり、number_of_clip_marks は、0 であってもよい。mark_typeは、マークのタイプを示す 8 ビットのフィールドであり、図 7 6 に示すテーブルに従って符号化される。

mark_time_stampは、32 ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図 7 7 に示すように、PlayList()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この 8 ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この 8 ビットのフィールドは何も意味を持たず、0 にセットされる。character_setの 8 ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図 1 9 に示される値に対応する。

name_lengthの 8 ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキ

ャラクター文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

MakersPrivateDataについては、図 2 2 を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ 1 つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、及び、それぞれのPlayListを代表する画像をストアする。全てのメニューサムネイルは、ただ 1 つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマークサムネイルは、ただ 1 つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は 1 つのtn_blockに格納される。1 つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1 つのサムネイル画像のバイト長は可変である。

図 7 8 は、menu.thmbとmark.thmbのシンタクスを示す図であり、図 7 9 は、図 7 8 に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図 7 9 に示したThumbnailのシンタクスについて説明すると、version_numberは、このThumbnail()のバージョンナンバを示す 4 個のキャラクター文字である。version_numberは、I S O 6 4 6 に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、及びClipMark()の中のref_thumbnail_indexによって参照される。

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図80に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I-picture)をとらなければならない。

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tn_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。

tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全てのtn_blockは、同じサイズ（固定長）であり、その大きさはtn_block_sizeによって定義される。

図8 1 A及び図8 1 Bは、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図8 1 A及び図8 1 Bのように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、“M2TS”ディレクトリ（図1 4）にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2 トランスポートストリームの構造は、図8 2に示すようになっている。AVストリームファイルは、DVR MPEG 2トランスポートストリームの構造を持つ。DVR MPEG 2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6 1 4 4 バイト（2 0 4 8 * 3 バイト）である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、1 9 2バイト長である。1つのソースパケットは、TP_extra_headerとトランスポートパケットからなる。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、1 8 8バイト長である。

1つのAligned unitは、3 2個のソースパケットからなる。DVR MPEG 2トランスポートストリームの中の最後のAligned unitも、また3 2個のソースパケットからなる。よって、DVR MPEG 2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が3 2の倍数でないとき、ヌルパケット（PID=

0xFFFFのトランスポートパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランスポートストリームに余分な情報を付加してはならない。

図83に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図83に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリーム又はパシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1又はISO/IEC13818-9に従っていなければならない。MPEG2トランスポートストリームのi番目のバイトは、TSTD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) とソースパケットタイザへ、時刻t(i)に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

27MHz PLL52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。arrival time clock counter53は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

source packetizer54は、全てのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がTSTDとソースパケットタイザの両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

2つの連続して入力されるトランスポートパケットの時間間隔が、230/2700

0000秒（約40秒）以上になる場合、その2つのトランスポートバケットのarrival_time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

smoothing buffer 55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファは、オーバーフローしてはならない。Rmaxは、スムージングバッファが空でないときのスムージングバッファからのソースバケットの出力ビットレートである。スムージングバッファが空であるとき、スムージングバッファからの出力ビットレートは0である。

次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = TS_recording_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptor等、において定義される値を参照してもよい。

smoothing buffer sizeは、入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファの大きさは0である。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptor等において定義される値を参照してもよい。

記録機（レコーダ）及び再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytes である。

次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについ

て説明する。図 8 4 は、D V R M P E G - 2 トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。D V R M P E G - 2 トランスポートストリームは、このモデルに従う。

2 7 M H z X - t a l 6 1 は、2 7 M H z の周波数を発生する。2 7 M H z 周波数の誤差範囲は、 ± 30 ppm (2 7 0 0 0 0 0 0 \pm 810 Hz) でなければならない。arrival time clock counter 6 2 は、2 7 M H z の周波数のパルスをカウントするバイナリカウンタである。Arrival_time_clock(i) は、時刻 $t(i)$ における Arrival time clock counter のカウント値である。

smoothing buffer 6 4 において、Rmax は、スムージングバッファがフルでないときのスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルであるとき、スムージングバッファへの入力ビットレートは 0 である。

M P E G - 2 トランスポートストリームの出力タイミングを説明すると、現在のソースパケットの arrival_time_stamp が arrival_time_clock(i) の LSB 3 0 ビットの値と等しいとき、そのソースパケットのトランスポートパケットは、スムージングバッファから引き抜かれる。Rpk は、トランスポートパケットレートの瞬時的な

最大値である。スムージングバッファは、アンダーフローしてはならない。

D V R M P E G - 2 トランスポートストリームのプレーヤモデルのパラメータについては、上述した D V R M P E G - 2 トランスポートストリームのレコーダモデルのパラメータと同一である。

図 8 5 は、Source packet のシンタクスを示す図である。transport_packet() は、ISO/IEC 13818-1 で規定される M P E G - 2 トランスポートパケットである。図 8 5 に示した Source packet のシンタクス内の TP_Extra_header のシンタクスを図 8 6 に示す。図 8 6 に示した TP_Extra_header のシンタクスについて説明すると、copy_permission_indicator は、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、又は copy prohibited とすることができる。図 8 7 は、copy_permission_indicator の値と、それらによって指定されるモードの関係を示す。

copy_permission_indicatorは、全てのトランスポートパケットに付加される。IEEE1394デジタルインタフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けてもよい。IEEE1394デジタルインタフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けてもよい。

arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

において、arrival_time_stampによって指定される値を持つ整数値である。

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS (Presentation Time Stamp) が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければ

ならず、且つ後述するDVR－STDに従わなければならない。

本実施例においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”

データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図88は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図88においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図88に示したTS1は、Clip1(Clip AVストリーム)の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータからなる。

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time(図88においてIN_time1で図示されている)に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケッ

トまでのストリームデータである。

また、図 8 8 における T S 2 は、Clip2 (Clip A V ストリーム) の影を付けられたストリームデータと Bridge-Clip の RSPN_arrival_time_discontinuity 以後の影を付けられたストリームデータからなる。T S 2 に含まれる Bridge-Clip の RSPN_arrival_time_discontinuity 以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuity で参照されるソースパケットから、Bridge-Clip の最後のソースパケットまでのストリームデータである。T S 2 の Clip2 の影を付けられたストリームデータは、RSPN_enter_to_current_Clip で参照されるソースパケットから、現在の PlayItem の OUT_time (図 8 8 において OUT_time2 で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図 8 9 は、Bridge-Clip を使用しない場合の先行する PlayItem と現在の PlayItem の関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図 8 9 における T S 1 は、Clip1 (Clip A V ストリーム) の影を付けられたストリームデータからなる。T S 1 の Clip1 の影を付けられたストリームデータは、先行する PlayItem の IN_time (図 8 9 において IN_time1 で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1 の最後のソースパケットまでのデータである。

また、図 8 9 における T S 2 は、Clip2 (Clip A V ストリーム) の影を付けられたストリームデータからなる。

T S 2 の Clip2 の影を付けられたストリームデータは、Clip2 の最初のソースパケットから始まり、現在の PlayItem の OUT_time (図 8 9 において OUT_time2 で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図 8 8 と図 8 9 において、T S 1 と T S 2 は、ソースパケットの連続したストリームである。次に、T S 1 と T S 2 のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、T S 1 と T S 2 の中に

含まれるプログラムの数は、1でなければならない。TS 1とTS 2の中に含まれるビデオストリームの数は、1でなければならない。TS 1とTS 2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS 1とTS 2の中に含まれるオーディオストリームの数は、等しくなければならない。TS 1及び/又はTS 2の中に、上記以外のエレメンタリーストリーム又はプライベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図90は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図90に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図91に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図90のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなる。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMP EG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなる。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、1つの連続でMP EG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならないが、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図90に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図92に示す。Clip1のビデオストリームは、図90のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなり、それは、1つの連続で

MPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなり、それは、1つの連続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明すると、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあってはならない。接続点において、トップボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、“top_field_first”及び“repeat_first_field”フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明すると、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例、MPEG1レイヤ2、AC-3、SESF L PCM、AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明すると、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャ

ップがあってはならない。図 9 3 に示すように、2 オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあってもよい。TS 2 のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述する DVR - STD に従わなくてはならない。

Clip 及び Bridge-Clip の制限について説明すると、TS 1 と TS 2 は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clip を使用する場合にのみ適用される。TS 1 の最後のソースパケットと TS 2 の最初のソースパケットの接続点においてのみ、Bridge-Clip AV ストリームは、ただ 1 つのアライバルタイムベースの不連続点を持つ。ClipInfo() において定義される RSPN_arrival_time_discontinuity が、その不連続点のアドレスを示し、それは TS 2 の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo() において定義される RSPN_exit_from_previous_Clip によって参照されるソースパケットは、Clip1 の中のどのソースパケットでもよい。それは、Aligned unit の境界である必要はない。BridgeSequenceInfo() において定義される RSPN_enter_to_current_Clip によって参照されるソースパケットは、Clip2 の中のどのソースパケットでもよい。それは、Aligned unit の境界である必要はない。

PlayItem の制限について説明すると、先行する PlayItem の OUT_time (図 8 8、図 8 9 において示される OUT_time1) は、TS 1 の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在の PlayItem の IN_time (図 8 8、図 8 9 において示される IN_time2) は、TS 2 の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clip を使用する場合のデータアロケーションの制限について、図 9 4 を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AV ストリームファイル) と Clip2 (Clip AV ストリームファイル) に接続される Bridge-Clip AV ストリームを、データアロケーション規定を満たすように配置す

ることによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip1 (Clip A Vストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip A Vストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip A Vストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図95を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip A Vストリームファイル) の最後の部分とClip2 (Clip A Vストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1 (Clip A Vストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip A Vストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

次に、DVR-ST Dについて説明する。DVR-ST Dは、DVR M P E G 2トランスポートストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-ST Dは、上述したシームレス接続された2つのPlayItemによって参照されるA Vストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルでもある。

DVR-ST Dモデルを図96に示す。図96に示したモデルには、DVR M P E G 2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On及びP n(k)の表記方法は、I S O / I E C 1 3 8 1 8 - 1 のT-ST Dに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリー

ムのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

DVR-S T Dのデコーディングプロセスについて説明する。単一のDVR M P E G - 2トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、I S O / I E C 13818-1に規定されているT-S T Dと同じである。復号動作と表示動作の規定もまた、I S O / I E C 13818-1に規定されているT-S T Dと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図88に示した）T S 1とT S 2の再生について説明する。T S 1は、先行するストリームであり、T S 2は、現在のストリームである。

図97は、あるAVストリーム（T S 1）からそれにシームレスに接続された次のAVストリーム（T S 2）へと移るときのトランスポートパケットの入力、

復号、表示のタイミングチャートを示す。所定のAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る間には、TS2のアライバルタイムベースの時間軸(図97においてATC2で示される)は、TS1のアライバルタイムベースの時間軸(図97においてATC1で示される)と同じでない。

また、TS2のシステムタイムベースの時間軸(図97においてSTC2で示される)は、TS1のシステムタイムベースの時間軸(図97においてSTC1で示される)と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパッケージがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパッケージのarrival_time_stampによって決定される。

TS1の残りのパッケージは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBn又はTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。したがって、時刻T1からT2までの区間では、ソースパッケージのarrival_time_stampは無視される。

N1をTS1の最後のビデオパッケージに続くTS1のトランスポートパッケージのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T2 - T1 = N1 / TS_recording_rate$$

(TS1)時刻T1乃至T2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

T2の時刻において、arrival time clock counterは、TS2の最初のソースパ

ケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TBn
又はTBsysのバッファへの入力タイミングは、TS2のソースケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについて説明すると、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるように、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明すると、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図97ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS2startは、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、Tppは、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC_delta = PTS1end + Tpp - PTS2start$$

オーディオのプレゼンテーションのタイミングについて説明すると、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である（図97に図示されている“audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

DVR-STDのシステムタイムクロックについて説明すると、時刻T5にお

いて、TS 1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていてもよい。この区間では、DVR-ST Dは、システムタイムクロックを古いタイムベースの値(STC 1)と新しいタイムベースの値(STC 2)の間で切り替える。STC 2の値は、次式により算出される。

$$STC2 = STC1 - STC_delta$$

バッファリングの連続性について説明する。STC11video_endは、TS 1の最後のビデオパケットの最後のバイトがDVR-ST DのTB1へ到着するときのシステムタイムベースSTC 1上のSTCの値である。STC22video_startは、TS 2の最初のビデオパケットの最初のバイトがDVR-ST DのTB1へ到着するときのシステムタイムベースSTC 2上のSTCの値である。STC21video_endは、STC11video_endの値をシステムタイムベースSTC 2上の値に換算した値である。STC21video_endは、次式により算出される。

$$STC21video_end = STC11video_end - STC_delta$$

DVR-ST Dに従うために、次の2つの条件を満たすことが要求される。まず、TS 2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC22video_start > STC21video_end + \Delta T1$$

この不等式が満たされるように、Clip 1及び、又は、Clip 2の部分的なストリームを再エンコード及び、又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC 1とSTC 2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS 1からのビデオパケットの入力とそれに続くTS 2からのビデオパケットの入力は、ビデオバッファをオーバーフロー及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、以上の例は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

次に、図98は、PlayListファイルの別の例を示す。図98と図23のシンタクスの大きな違いは、UIAppInfoPlayList()をストアしている場所である。図98の例では、UIAppInfoPlayList()がPlayList()の中から外に出されているので、UIAppInfoPlayList()の将来の情報拡張が比較的容易に行えるようになる。

version_numberは、このサムネイルヘッダ情報ファイルのバージョンナンバを示す4個の数字である。

PlayList_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

PlayListMark_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

図99は、図98のPlayListファイルの中のUIAppInfoPlayListのシンタクスを示す。PlayList_service_typeは、PlayListファイルのタイプを示す。その一例は、図26に示されている。また、PlayList_service_typeは、デジタルTV放送のプログラムが示すサービスタイプと同じ意味を持たせてもよい。例えば、日本のデジタルBS放送の場合、サービスタイプは、テレビサービス、音声サービス、及びデータ放送サービスの3種類を持つ。PlayListが使用するClip AVストリームが含むプログラムのサービスタイプを代表する値をPlayList_service_typeにセットする。

PlayList_character_setは、channel_name, PlayList_name及びPlayList_detailフィールドに符号化されているキャラクター文字の符号化方法を示す。また、

これはPlayListMarkの中のmark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。

channel_numberは、そのPlayListが記録されるとき、ユーザによって選択された放送チャンネル番号又はサービス番号を示す。複数のPlayListが1つのPlayListにコンバインされた場合は、このフィールドはそのPlayListの代表値を示す。このフィールドが0xFFFFにセットされている場合、このフィールドは何も意味を持たない。

channel_name_lengthは、channel_nameフィールドの中に示されるチャンネル名のバイト長を示す。このフィールドは、20以下の値である。

channel_nameは、そのPlayListが記録されるとき、ユーザによって選択された放送チャンネル又はサービスの名前を示す。このフィールドの中の左からchannel_name_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。複数のPlayListが1つのPlayListにコンバインされた場合は、このフィールドはそのPlayListを代表する名前を示す。

PlayList_name_lengthは、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。

PlayList_nameは、PlayListの名前を示す。このフィールドの中の左からPlayList_name_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。

PlayList_detail_lengthは、PlayList_detailフィールドの中に示されるPlayListの詳細情報のバイト長を示す。このフィールドは、1200以下の値である。

PlayList_detailは、PlayListの詳細情報を説明するテキストを示す。このフィールドの中の左からPlayList_detail_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記テキストを示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。

これ以外のシンタクスフィールドの意味は、図 2 7 に示す同名のフィールドと同じである。

図 1 0 0 は、図 9 8 の PlayList ファイルの中の PlayList() のシンタクスを示す。図 2 5 の例と比べると、UIAppInfoPlayList() がなくなった点が違うだけで、これ以外は基本的に同じである。

図 1 0 1 は、SubPlayItem のシンタクスの別例を示す。図 4 0 の例と比べると、STC_sequence_id が追加された点が大きな違いである。

STC_sequence_id は、Clip_Information_file_name に対応する A V ストリームファイル上の再生区間を特定するための SubPath_IN_time と SubPath_OUT_time が参照するところの S T C の STC_sequence_id を示す。SubPath_IN_time と SubPath_OUT_time は、STC_sequence_id によって指定される同じ S T C 連続区間上の時間を示す。

SubPlayItem に STC_sequence_id を追加することにより、SubPlayItem が参照する A V ストリームファイルが S T C 不連続点を持つことが許されるようになる。

これ以外のシンタクスフィールドの意味は、図 4 0 に示す同名のフィールドと同じである。

図 1 0 2 は、Real PlayList の作成方法を説明するフローチャートを示す。図 1 の記録再生装置のブロック図を参照しながら説明する。

ステップ S 1 1 で、制御部 2 3 は Clip A V ストリームを記録する。

ステップ S 1 2 で、制御部 2 3 は Clip A V ストリームの EP_map を作成可能どうかを調べる。ステップ S 1 2 で、Y e s の場合はステップ S 1 3 へ進み、EP_map を作成する。ステップ S 1 2 で、N o の場合はステップ S 1 4 へ進み、TU_map を作成する。

その後、ステップ S 1 5 で、制御部 2 3 は PlayList の CPI_type をセットする。

ステップ S 1 6 で、制御部 2 3 は上記 Clip の全ての再生可能範囲をカバーする PlayItem からなる PlayList() を作成する。CPI_type が EP_map タイプの場合は、時間情報を P T S ベースでセットする、このとき、Clip の中に S T C 不連続点があり、PlayList() が 2 つ以上の PlayItem からなる場合は、PlayItem 間の connection_condition もまた決定する。CPI_type が TU_map タイプの場合は、時間情報をアライバルタイムベースでセットする。

ステップS 17で、制御部23はUIAppInfoPlayList()を作成する。

ステップS 18で、制御部23はPlayListMarkを作成する。

ステップS 19で、制御部23はMakersPrivateDataを作成する。

ステップS 20で、制御部23はReal PlayListファイルを記録する。

このようにして、新規にClip AVストリームを記録する毎に、1つのReal PlayListファイルが作られる。

図103は、Virtual PlayListの作成方法を説明するフローチャートである。

ステップS 31で、ユーザインタフェースを通して、ディスクに記録されている1つのReal PlayListが指定される。そして、そのReal PlayListの再生範囲の中から、ユーザインタフェースを通して、IN点とOUT点で示される再生区間が指定される。CPI_typeがEP_mapタイプの場合は、再生区間をPTSベースでセットし、CPI_typeがTU_mapタイプの場合は、再生区間をアライバルタイムベースでセットする。

ステップS 32で、制御部23はユーザによる再生範囲の指定操作が全て終了したか調べる。ユーザが上記指示した再生区間に続けて再生する区間を選ぶ場合はステップS 31へ戻る。ステップS 32でユーザによる再生範囲の指定操作が全て終了した場合は、ステップS 33へ進む。

ステップS 33で、連続して再生される2つの再生区間の間の接続状態(connection_condition)を、ユーザがユーザインタフェースを通して決定するか、又は制御部23が決定する。

ステップS 34で、CPI_typeがEP_mapタイプの場合、ユーザインタフェースを通して、ユーザがサブバス(アフレコ用オーディオ)情報を指定する。ユーザがサブバスを作成しない場合はこのステップはない。

ステップS 35で、制御部23はユーザが指定した再生範囲情報、及びconnection_conditionに基づいて、PlayList()を作成する。

ステップS 36で、制御部23はUIAppInfoPlayList()を作成する。

ステップS 37で、制御部23はPlayListMarkを作成する。

ステップS 38で、制御部23はMakersPrivateDataを作成する。

ステップS 39で、制御部23はVirtual PlayListファイルを記録する。

このようにして、ディスクに記録されているReal Playlistの再生範囲の中から、ユーザが見たい再生区間を選択してその再生区間をグループ化したものに、1つのVirtual Playlistファイルが作られる。

図104はPlaylistの再生方法を説明するフローチャートである。

ステップS51で、制御部23はInfo.dvr, Clip Information file, Playlist file及びサムネイルファイルの情報を取得し、ディスクに記録されているPlaylistの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS52で、制御部23はそれぞれのPlaylistのUIAppInfoPlaylist()に基づいて、Playlistを説明する情報をGUI画面に提示する。

ステップS53で、ユーザインタフェースを通して、GUI画面上からユーザが1つのPlaylistの再生を指示する。

ステップS54で、制御部23は、CPI_typeがEP_mapタイプの場合、現在のPlayItemのSTC-sequenc-idとIN_timeのPTSから、IN_timeより時間的に前で最も近いエントリポイントのあるソースパケット番号を取得する。又は制御部23は、CPI_typeがTU_mapタイプの場合、現在のPlayItemのIN_timeから、IN_timeより時間的に前で最も近いタイムユニットの開始するソースパケット番号を取得する。

ステップS55で、制御部23は上記ステップで得られたソースパケット番号からAVストリームのデータを読み出し、AVデコーダ27へ供給する。

ステップS56で、現在のPlayItemの時間的に前のPlayItemがあった場合は、制御部23は、前のPlayItemと現在のPlayItemとの表示の接続処理をconnection_conditionに従って行う。

ステップS57で、制御部23は、CPI_typeがEP_mapタイプの場合、AVデコーダ27は、IN_timeのPTSのピクチャから表示を開始するように指示する。又は、制御部23は、CPI_typeがEP_mapタイプの場合、AVデコーダ27は、IN_time以後のストリームのピクチャから表示を開始するように指示する。

ステップS58で、制御部23は、AVデコーダ27にAVストリームのデコードを続けるように指示する。

ステップS59で、制御部23は、CPI_typeがEP_mapタイプの場合、現在表示

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.